

Rapport 6eme séance : 7/02/2022

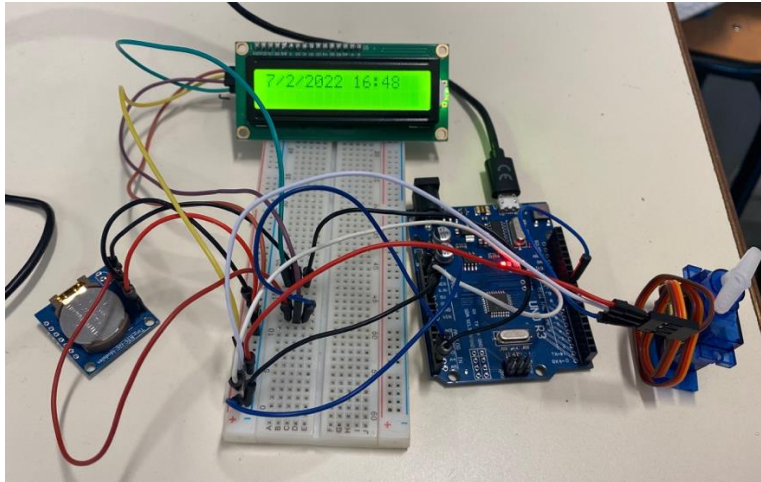
Projet : Med Dispenser

1 ère partie

Avant le cours et pendant le début de la séance, j'ai essayé d'afficher un texte sur l'écran en utilisant l'afficheur LCD I2C. Mais ça ne marchait pas j'ai donc essayé plusieurs montages différents et plusieurs codes mais je n'y arrivais pas. Finalement le problème était le contraste : l'écran affichait ce que je voulais mais le contraste était trop grand on ne pouvait donc pas voir le texte.

Après avoir modifié le contraste, j'ai remplacé l'ancien code de l'écran avec le nouveau dans le code parent pour pouvoir afficher l'heure.

Montage et code



```
codeparent  horlogeecran  servomoteur_
#include <Wire.h>
#include <RTCLib.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
//RTC_DS1307 RTC;

void initHour() {
  Serial.begin(57600);
  lcd.init();
  Wire.begin();
  RTC.begin();
  lcd.begin(16,2);
  lcd.setCursor(0,0);
}

void printHour() {
  lcd.backlight();
  DateTime now=RTC.now();
  lcd.print(now.day(), DEC);
  lcd.print("/");
  lcd.print(now.month(), DEC);
  lcd.print("/");
  lcd.print(now.year(), DEC);
  lcd.print(" ");
  lcd.print(now.hour(), DEC);
  lcd.print(":");
  lcd.print(now.minute(), DEC);
  delay(1000);
  lcd.clear();
}
}
```

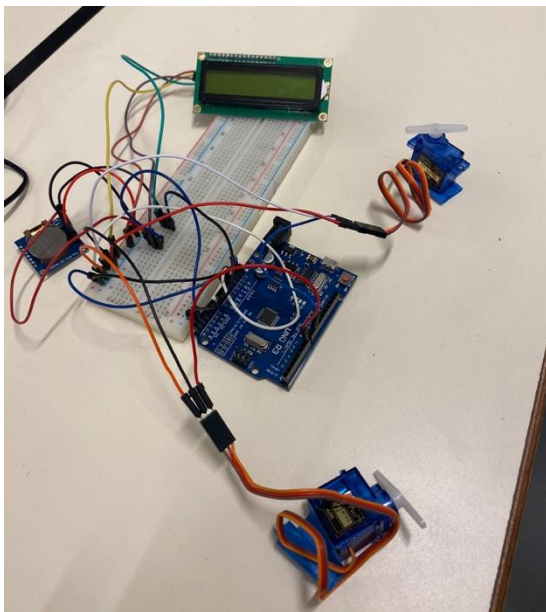
2^{ème} partie

Durant la suite de la séance j'ai vérifié si le moteur s'activait à la bonne heure avec le nouvel afficheur. Cela fonctionne mais l'écran n'affiche rien quand le moteur tourne. J'essaierai de régler ce problème plus tard si j'ai le temps.

Ensuite j'ai voulu faire en sorte de pouvoir choisir combien de médicaments on peut faire tomber. La seule solution que j'ai trouvée est d'activer le moteur une seule fois par minute en utilisant un delay(). Puis on actionnera le moteur le nombre de fois qu'il faudra. Le problème de cette méthode est qu'il faut plusieurs minutes pour faire tomber plusieurs médicaments.

Puis j'ai connecté le deuxième moteur, j'ai d'abord fait le montage puis j'ai copier-coller le code en faisant des changements. Mais ça ne fonctionne pas : seul un moteur se met en marche. J'ai donc inversé les moteurs pour être sûre que ce ne soit pas un défaut du moteur, ce qui n'est pas le cas. Pour la prochaine fois je dois donc résoudre ce problème.

Montage et code



```
codeparent horlogeecran servomoteur_1 servomoteur_2
//code principal
#include <RTClib.h>
RTC_DS1307 RTC;

void setup() {
  // put your setup code here, to run once:
  initHour();
  initMotor1();
  initMotor2();
}

void loop() {
  // put your main code here, to run repeatedly:
  printHour();
  //delay(3000);
  DateTime now=RTC.now();
  /*if ((now.hour()==17) && (now.minute()==52)){
    runMotor1();
    runMotor2();
    delay(60000);}*/
  runMotor1();
  runMotor2();
}
```

codeparent	horlogeecran	servomoteur_1	servomoteur_2
------------	--------------	---------------	---------------

```
#include <Servo.h>
Servo monServo1;
int position;

void initMotor1() {
  monServo1.attach(9);
  position=0;
}

void runMotor1() {

  // Fait bouger le bras de 0° à 180°
  for (int position = 0; position <= 180; position++) {
    monServo1.write(position);
    delay(15);
  }

  // Fait bouger le bras de 180° à 10°
  for (int position = 180; position >= 0; position--) {
    monServo1.write(position);
    delay(15);
  }
}
```

codeparent	horlogeecran	servomoteur_1	servomoteur_2
------------	--------------	---------------	---------------

```
#include <Servo.h>
Servo monServo2;
int pos;

void initMotor2() {
  monServo1.attach(9);
  pos=180;
}

void runMotor2() {

  // Fait bouger le bras de 180° à 10°
  for (int pos = 180; pos >= 0; pos--) {
    monServo2.write(pos);
    delay(15);
  }

  // Fait bouger le bras de 0° à 180°
  for (int pos = 0; pos <= 180; pos++) {
    monServo2.write(pos);
    delay(15);
  }
}
```